Minds and Machines
https://doi.org/10.1007/s11023-018-9484-3

Peeking Inside the Black Box: A New Kind of Scientific Visualization

Michael T. Stuart[1] · **Nancy J. Nersessian**[2]

Received: 28 April 2018 / Accepted: 12 November 2018

Abstract
Computational systems biologists create and manipulate computational models of biological systems, but they do not always have straightforward epistemic access to the content and behavioural profile of such models because of their length, coding idiosyncrasies, and formal complexity. This creates difficulties both for modellers in their research groups and for their bioscience collaborators who rely on these models. In this paper we introduce a new kind of visualization (observed in a qualitative study of a systems biology laboratory) that was developed to address just this sort of epistemic opacity. The visualization is unusual in that it depicts the dynamics and structure of a computer model instead of that model's target system, and because it is generated algorithmically. Using considerations from epistemology and aesthetics, we explore how this new kind of visualization increases scientific understanding of the content and function of computer models in systems biology to reduce epistemic opacity.

Keywords Computer simulation · Epistemology of computer simulation · Epistemic opacity · Exemplification · Epistemology of photography · Model-based reasoning · Scientific understanding · Scientific visualization

1 Introduction

Computational systems biology is an interdisciplinary field that uses computational methods to address questions about complex biological systems that are not currently answerable using wet lab experimentation. When applied to model the dynamics of complex biological systems, computational systems biology has enjoyed both predictive and explanatory success. However the models themselves

✉ Michael T. Stuart
mike.stuart.post@gmail.com

[1] Department of Philosophy, University of Geneva, Rue de Candolle 2, 1211 Geneva, Switzerland

[2] Department of Psychology, Harvard University, William James Hall, 33 Kirkland St., Cambridge, MA 02138, USA

Published online: 26 November 2018

are "epistemically opaque," not just in the sense that no human could verify all their inferences (Humphreys 2004; Lenhard 2018), but also in the sense that their formal complexity, long length, and idiosyncrasies in coding make it very difficult for others—modellers or biologists—to grasp their content and behavioural profile. This makes the models difficult to understand, interpret, and trust.

In a qualitative study of a computational systems biology laboratory, we found scientists tackling this version of epistemic opacity by creating visualizations. In this paper, we present a novel kind of scientific visualization that was developed by the lab, which as far as we know has not appeared in any philosophical literature. The visualization is novel in that it is the automatic output of a program designed to generate diagrams of the inner workings of computer models. In other words, the output visualization is a representation of the *computer model*, rather than the biological system being modelled. This kind of visualization is able to resolve the lab's specific problem of epistemic opacity, in part because the diagram simplifies and draws attention to important parts of the model. But the fact that it is algorithmically generated is also epistemologically interesting. Usually, such model-diagrams are drawn by hand (on computer), and there is no way to verify how well the diagram captures the dynamics of the model. We must simply trust its creator. In the case to be discussed, we can verify that the algorithm is producing a diagram that accurately represents the model when it is run on simple, well-understood models. When it is used on more complex models, the output representations can be checked against empirical data (see Sect. 3). Together, these checks justify scientific confidence in the accuracy of such algorithmically produced visual representations.

We begin our philosophical analysis in Sect. 4 by noting that the visualization appears to work as an exemplar (Elgin 2011). Exemplars represent some features of a target system, but they also instantiate those same features. For example, a sample piece of fabric represents a much larger, unseen piece of fabric as having certain features, e.g., having a certain colour and texture. But it also instantiates those same features, because the sample piece actually possesses the colour and texture of the fabric that it represents.

Because an exemplar instantiates features of a target, it provides the user with access to those features. This is epistemologically relevant because access to features of interest are necessary (if not sufficient) for some kinds of knowledge and understanding. For example, if I want to help someone understand why they should pick fabric x over fabric y, I can give them a sample of both and let them see that x has more desirable qualities than y. Exemplars are therefore useful when trying to help someone increase her/his understanding by giving them access to the features of a system that we know are instantiated in both exemplar and target. In many scientific cases, however, the exemplar "comes first," in the sense that we have an exemplar which *purports* to instantiate and refer to features of a target system, yet our lack of independent access to the target system itself prevents us from confirming that such features are indeed instantiated in the target as represented by the exemplar.

In such cases, there can be good reasons for believing that exemplified features are indeed as the exemplar represents them to be in the target system, even without direct independent verification. This is achieved in the case to be discussed by drawing attention to the fact that the instantiation of certain features in the exemplar is

counterfactually dependent on those same features being instantiated in the target system. The visualization to be discussed is produced in such a way that if the model did not instantiate those features, they would not be instantiated in the visualization. But they do appear in the visualization. This opens our access to (and increases our understanding of) the features of the computer model we are interested in.

This is our second epistemological consideration: counterfactual dependence justifies our use of the visualization as a guide to the computer model. Both photographs and paintings can exemplify, that is, instantiate and represent features of their targets, but because of the (more or less) direct causal dependence of photographs on their targets (Walton 1984, 2013), this kind of exemplar generally provides greater epistemic access to certain (e.g., visual) features of the target. As the algorithmically generated visualizations to be discussed are strongly counterfactually dependent on the models they exemplify, they can be taken as trustworthy guides to those models. Through this, the computer-generated visualizations can increase our understanding of the models they depict.

In the next section we give some background on the laboratory we studied and the specific version of epistemic opacity that the researchers were trying to address. Then we present the visualization that was created to address it (Sect. 3), followed by a discussion of its epistemology in Sect. 4.

2 Epistemic Opacity In Vivo

The integration of computational and traditional scientific methods has been a driving force for progress in many scientific fields, as computational methods extend the reach of our hands and minds. But they come at an epistemological price: "no human can examine and justify every computational step performed by the computer, because the steps are too numerous" (Parker 2014, 142). Normally, to justify a logical inference, we check that each step is justified. However, because of the number of steps in most computer simulations, this is not possible (Humphreys 2004, 2009). Thus, "a process is essentially epistemically opaque to X if and only if it is impossible, given the nature of X [e.g., as a limited cognitive agent], for X to know all of the epistemically relevant elements of the process" (Humphreys 2009, 618).

The problem of epistemic opacity is often framed in terms of knowledge. The lack of access to all the steps of a computer simulation means we cannot ensure its conclusion is justified (at least, not in the usual way), because we cannot verify all the premises and subderivations. And if we cannot justify the output, we cannot know it, insofar as knowledge requires justification. This is a problem that systems biologists recognize, but it is not the main problem related to the epistemic opacity of computer models. Lacking access to all of the components and steps of

a computer model frustrates their ability to explain or predict the output of models, and to grasp their empirical content. These goals are best characterized as goals of *understanding* rather than knowledge. Understanding has as yet no accepted philosophical definition, but there are at least three main brands on offer. Explanatory accounts identify understanding with the possession of explanations.[1] Manipulation accounts locate understanding in the abilities of agents who understand.[2] Objectual accounts locate understanding in the grasping of "coherence-making" relations between parts of a network of elements.[3] In all cases, complete understanding cannot be achieved if our access to the object of understanding is incomplete.

To see why this is so, we need to say something more about access. There are different kinds of access we can use to gain understanding, including cognitive, causal, empirical, and computational. Consider a case where we gain computational access but lack cognitive access. It has been proven that only four colours are necessary to colour any map of any possible arrangement of countries so that two countries sharing a border never have the same colour. This is called "the Four Colour Theorem" (see, e.g., Wilson 2014). The proofs that three colours are not enough and that five are enough were completed by human mathematicians. But answering the question of whether four colours were enough required a supercomputer. In this case, the computer provided access to this truth about a general property of maps that no human cognitive agent could. Further, through the computer we gain computational access to the justificatory grounds of the claim that only four colours are needed (because the computer has access to all the steps relevant for justification). We lack a comparable cognitive access, because no human could observe (or otherwise cognitively process) all the steps performed by the computer that were relevant to justify the inference in a single lifetime. Without such cognitive access, no individual human mind can "possess" the justification for why only four colours are needed, and so no human can possess the full *explanation* for why only four colours are needed. Therefore no human can have complete understanding in the sense of explanatory understanding. (Of course, we could explain why only four colours are needed by saying "the machine said so." But this is very shallow understanding). Similar considerations apply for manipulability and objectual accounts of understanding: Our lack of cognitive access prevents us from gaining new abilities to manipulate, reproduce, explain, or justify the proof as a whole. We might gain abilities to manipulate (and thus understand) *parts* of the proof, but this is not what is in question. Finally, our lack of cognitive access to the proof prevents our grasping the relevant "coherence-making" relationships between the steps of the proof because without cognitive access to all the steps of the proof, we cannot grasp the relevant (semantic, deductive, inferential, explanatory, etc.) connections between them. We

[1] See Pritchard (2010, 74), Hempel (1965, 334), Kitcher (1989, 419), Grimm (2008) and De Regt (2009, 588), Khalifa (2012), Strevens (2013), Hills (2015) and Hannon (forthcoming).

[2] See Lenhard (2006), Stuart (2016, 2018), Wilkenfeld (2013, 2014, 2017) and Wilkenfeld and Hellmann (2014).

[3] See Baumberger (2011), Baumberger and Brun (2016), Dellsén (2018), Elgin (2007), Kvanvig (2009), Khalifa (2013), Wilkenfeld (2014) and Kelp (2015).

do not therefore have full understanding of the proof in the sense of objectual understanding either.

The example of the Four Colour Theorem suggests that a lack of cognitive access can frustrate all three senses of understanding. It also shows that computational access does not substitute for cognitive access. Whether causal, empirical, or some other kinds of access are also necessary, we do not say. Our point is merely that a complete lack of cognitive access will always prevent us from possessing explanations of the phenomenon in question, from gaining the skills to successfully manipulate the phenomenon in question, and from grasping connections relevant for understanding between aspects of the phenomenon in question. And this lack of access is what is at issue in the computational systems biology lab that we studied. The computer models they use are written in code that is idiosyncratically coded and so long that it is practically impossible for modellers and other scientists to possess the relevant explanations, gain the relevant abilities, or grasp the relevant connections among the parts of the models to secure understanding. Because of this, lab members and collaborators are not able to fully understand their own models and even less so the ones they do not write themselves.

This opacity of understanding is the kind of opacity that the novel visualization aims to assuage. To show this, we will first give some background on the laboratory we studied, and then present the problem in the lab members' own words.

One major goal of systems biology is to discover "how cells compute"; that is, how cells "make decisions" about what to do given the state of their environment (P1, presentation, 02/02/2016). This is a completely general question that could be asked of any given cell. As *systems* biologists, this lab would like to see "a catalogue of the different mechanisms that are at play, [such that we could] look at a particular cell and say what the mechanisms are going to be, and predict how the cell is going to behave" (P1, interview, 02/04/2016). This could be achieved by representing all the signalling pathways in the cell, which are pathways along which information travels between the environment, cells, and cell components. Through them, the basic activities of the cell, e.g., development, repair and death, are completed. Computational systems biologists often focus on signalling networks, which are large combinations of signalling pathways. The (distant) end goal is to create a library of models that includes worked out signalling networks for all cell types. The lab's Principal Investigator (P1) puts it this way, "We could actually do this, we could actually make these models, once and for all—basically they describe what we know about biochemistry—and put them in a large library and make models based on what we know," but, "different methods obtain different answers," and the methods for coupling their models with the models of others and with the huge amounts of data currently spewing from experimental biology laboratories "are still not really well worked out...Putting it all together into one thing, one piece of software, or one set of software tools that you could use to do this modelling; that's not happened yet. Not by a long shot" (P1, interview, 02/04/2016).

In the meantime, the lab has two proximate goals, both of which concern models. The first is to build models of cell components (usually signalling networks) that are behaviourally and predictively accurate. To do this, the model must be properly "grounded in the data" (P1, interview, 03/03/2016). They have to be behaviourally

and predictively accurate because they are also in the service of answering specific biological and medical questions, which in the case of this laboratory, usually come from an external collaborator. Of course, the fact that some of the modelling problems are dictated by external collaborators does not in any way frustrate the general goal of creating complete working models of cells.

The second goal naturally complements the first, since there would be no point in creating behaviourally and predictively accurate models of cells and cell components if no one could understand them. The models must not only predict what cells will do, they must also help us to understand how and why the cell does what it does in a way that biologists can access, use, and communicate. This is especially important in the context of systems biology, which is an interdisciplinary field whose practitioners usually divide into modellers with little understanding or experience with wet lab experimentation, and bioscientists, who are not typically comfortable with computer modelling (MacLeod and Nersessian 2016).

P1 puts it this way,

> We're still very much in the phase of trying to come up with ways to make the models accessible for interdisciplinary research, both to computational researchers and modellers, but also to biologists. So a lot of the work has been recently emphasizing how we make models understandable to everyone… If you have a bunch of code that describes a model, okay, it's nice that we can write the lines, we have code that can give you a precise model definition. [But] you have to write a computer code for each one, and it's messy and you can never figure out what's in it. Having a [programming] language allows you to standardize that to some extent. But it's still code, and it's still really hard to understand, even for people who are working in the same model. Especially for me as the PI of a group, I have students developing those models, I cannot go through their code line by line and figure out what's in every model because so much of it is understanding how it all fits together, which is really hard to work out, from just looking at lines of code (P1, interview, 02/04/2016).

P4 was a recent Ph.D. student who was leaving for a postdoctoral position during our time in the lab. He expressed the same concern with respect to collaboration: "There's this tendency in this field where modelling is treated like a black box. So if I, as a modeller, present a model, there's not much rigour in terms of [a collaborator] asking me, what's in the model? They just take my word for it. And that's bad, obviously." What is needed "is a step towards being more open about what's in the model." This could "bridge the communities" of modellers and experimentalists (P4, interview, 02/22/2016).

A natural way for computational systems biologists to do this employs visualizations. "Visualizing biochemical interactions has a long history of being conveyed through symbolic, pictorial and graphical representations," specifically, pathway diagrams representing metabolic and signaling processes. And so, "anytime you presented a model, you had to build a diagram" (P4, interview, 02/22/2016). This is because "there's something about the visual way of showing something that is very powerful. It speaks to an intuition that is not necessarily strictly defined, but it is a very powerful intuition nevertheless. There have been a lot of successes in biology,

with people thinking in this visual way, right? And so we want people with that intuition to be able to relate to models and modelling, and to be able to use models and understand what's in models." Even if a model is a black box, computationally speaking, it "shouldn't be treated as a black box, because if a collaboration should work, everybody should know what's in it. I think a visualization tool goes a great, long way towards not treating a model like a black box." The right visualization could help collaborators "see what's in it, without having to read the whole thing" (P4, interview, 02/22/2016).

The use of visualization is so attractive that it can seem like the only viable path to increasing understanding. Thus three lab members write: "Visual representations are *necessary* to understand individual rules as well as analyze interactions of hundreds of rules, which motivates the need for automated diagramming tools" (our emphasis, publication preprint, accessed 09/09/2016). But what kind of visualizations should be used? Given the central role of pathway diagrams in biological reasoning, it has been natural to visualize models using this kind of diagram. Producing diagrams that use the same visual conventions that bioscientists are used to should facilitate collaborator understanding of how the model produces its results.

This brings us to an important epistemological point. Until now, when a computational systems biologist presented the results of their work accompanied by a visualization, that visualization was always drawn by hand. It was always a human-made interpretation of the structure of that model. This opens the door to a certain kind of skepticism, as "there's no relationship to the model. I could give you a crap model, and say this is the diagram, and you will just have to believe me. There's no relationship to the actual code" (P4, interview, 03/01/2016). To avoid this kind of situation, the lab wanted to create an open-source visualization tool that would automatically produce a visualization for any computer model (written in a specific open-source coding language) that could be formally and empirically justified. This would alleviate skepticism about the accuracy of model visualizations, which are seen as the only serious contender for reducing epistemic opacity.

Given this situation, P1 told us that developing algorithmically produced visual representations of the models "is really one of the core things that we're working on, right now" (interview, 02/04/2016). It is "important for us, it helps us communicate a lot better, it makes our work more exciting, it makes it easier for us to read our own work" (P1, interview, 02/04/2016). Here is how they achieved this.

3 From Diagrams to Models to Hairballs (and Back)

The inputs and outputs of computational systems biology research are often pathway diagrams. At the beginning of a project, they are typically given a pathway diagram by their collaborator that encodes what the experimental collaborator knows, or is interested in, about a given system. These pathway diagrams are most often partial and insufficient for building a computer model of the system. The modeller needs to build the model from the diagram through searching the experimental literature and databases for what is known more broadly about the system (binding affinities, reaction speeds, etc.), in an iterative processes of

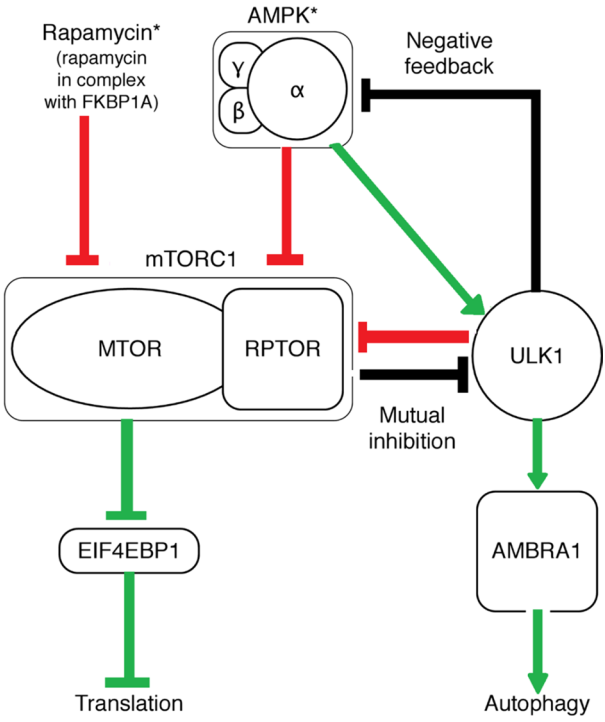

Fig. 1 Pathway diagram of Rapamycin's effects, drawn by hand (on a computer) by Szymańska et al. (2015)

building the model (Chandrasekharan and Nersessian 2015). When complete, they return new diagrams to their collaborators that represent and summarize the results of their modelling. Even though a computational modeller will not be able to go through and understand all the code of someone else's model, they want to feel confident that everything produced by the model is the result of transparent code, including the final diagram. Here is how the lab's new algorithm accomplished this, using a particular example.

Figure 1 presents a diagram of a biological model of a signalling pathway. Rapamycin is a drug that can initiate autophagy in a cell, which is the process by which a cell deconstructs itself. Autophagy is important because it prevents toxic elements of the cell's interior from spilling into the surrounding environment on cell death. It does this by deconstructing those elements so that they can be employed by other cells (see e.g., Mizushima and Komatsu 2011). It would be useful to create a drug that would initiate this process in a given cell, so that certain unwanted cells, e.g., cancer cells, could be targeted. These cells would then deconstruct themselves without posing any threat to nearby cells. Figure 1 presents our knowledge of the system gained through experimental manipulation.

There are a number of conventions that help us understand the diagram. Green arrows represent activation (usually by phosphorylation), while red blunted arrows represent inhibition. Two red blunted arrows in a row become green because two

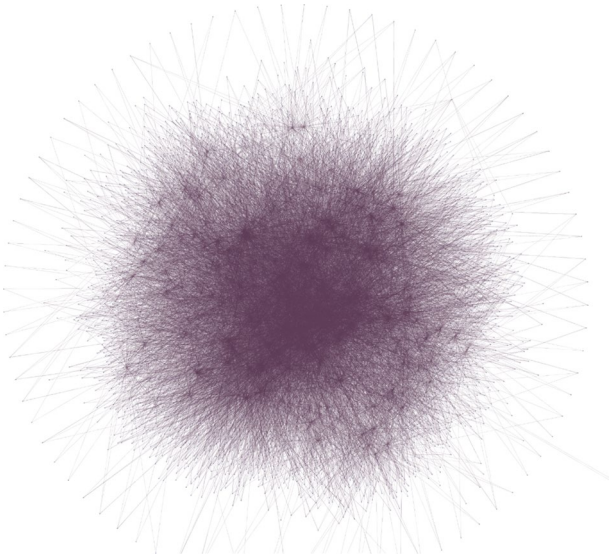

Fig. 2 Reaction network of the computational Rapamycin model

inhibitions have the same effect as an activation. A black blunted arrow represents negative feedback. A biologist looking at this diagram for the first time would be aware of most of the molecules by name, and would know the roles played by each of them in the cell. They would therefore quickly see how Rapamycin leads to autophagy by inhibiting mTORC1 (mTORC1 inhibits ULK1, but with the addition of Rapamycin this inhibition ceases, which allows ULK1 to activate AMBRA1, which leads to autophagy).

The lab created a computer model of this signalling network. The model is composed of sets of rules that govern transformations between states in the model, in a way that is meant to accurately predict what happens in real cells. The model's rules can be converted into a visual "reaction network," as displayed in Fig. 2.

The model contains 7 molecule types, 31 rules, and 6581 possible reactions, all of which are visualized here in Fig. 2. This is not a visualization of the signalling network, but of the reactions that can take place in the computational model of the signalling network. That is, it is a representation of the modal space of the computational model, not of the real-world target system.

P4 describes Fig. 2 as a "hairball." While it captures the full set of interactions possible within the model, it is impossible to see what is going on. It tells us nothing about the contents or dynamic structure of the model. Cognitively speaking, it is no more digestible than a hairball. As P4 puts it, "Biological understanding exists at a particular resolution…Biologists don't think at that [hairball] level, at that size, they think at a much coarser level than that" (P4, interview, 02/22/2016).

To address this, P4 in collaboration with P1 and other lab members, built a program that would take the above reaction network and distill it via a number of steps into a comprehensible visualization that would expose "the guts of the

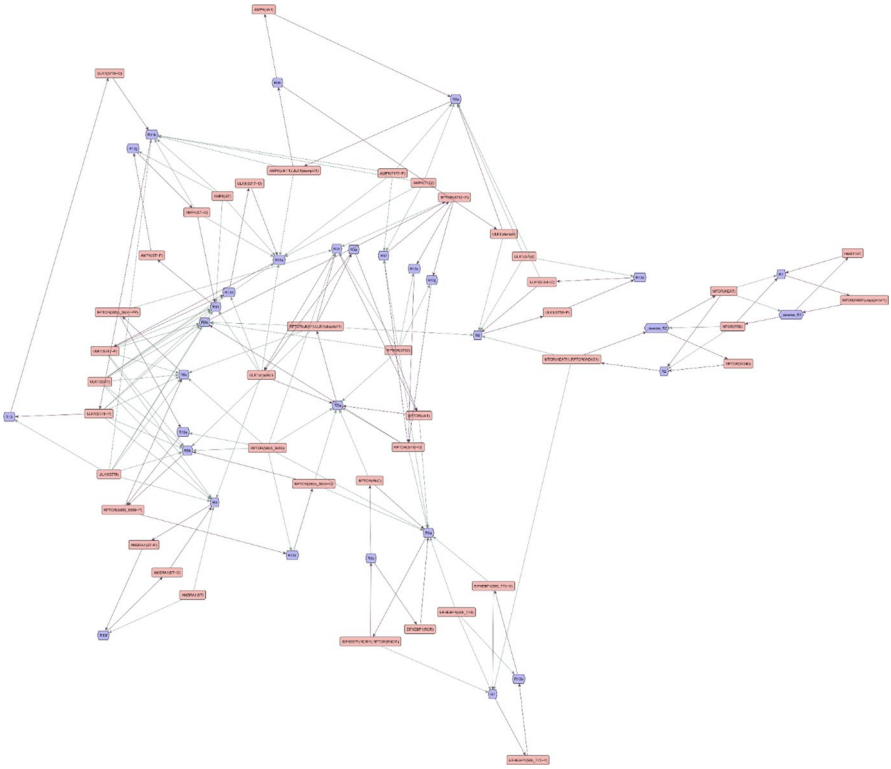

Fig. 3 Regulatory graph of the computational Rapamycin model

model" (P4, interview, 02/22/2016). The first step in this process is to create a "regulatory graph," as seen in Fig. 3.

This graph has the same molecule types and rules as the reaction network, but not all possible reactions are shown. Instead, the program creates a number of "regulatory relationships" that capture the overall behaviour of the rules that comprise the model. At the same time, the reaction network is converted into a graph, in which the nodes and edges of Fig. 3 replace the reactions of Fig. 2. The nodes represent states or rules, and the edges represent "influences" that transform those states/rules into other states/rules. Information is lost in the transfer from Figs. 2 to 3 only in the sense that individual reactions are no longer pictured. But that information is retrievable for the program, which can be run backwards to recreate the reaction network from the regulatory graph.

There is still a problem with this diagram, however, which is that there are too many nodes and edges. For a diagram like this to express a kind of overall "flow" among elements that is graspable by a human mind, there have to be less elements.

Consequently, the graph must be simplified. This is achieved through three processes: "pruning," "grouping," and "collapsing." In the pruning phase, a number of nodes and edges in the graph are chosen by the program to be foregrounded as

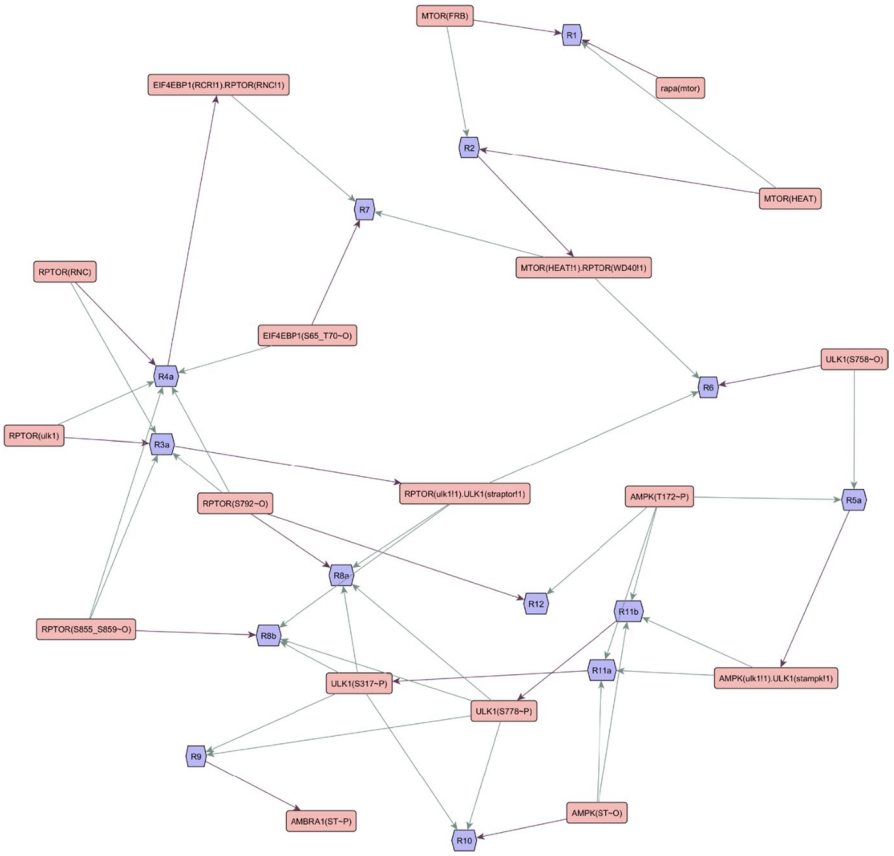

Fig. 4 "Pruned" regulatory graph with background nodes eliminated

significant by reducing the number of redundant or unnecessary nodes, called "background nodes." Background nodes are defined mathematically, but intuitively, they are nodes that have little influence on the overall function of the model. They might be free binding sites that are nowhere activated in the model, or redundant bindings that are already accounted for somewhere else. The definition of a background node can be altered or overridden by the user if she or he wants to include more or less background nodes. From 31 process nodes (hexagons), 42 state nodes (rectangles) and 161 edges (arrows), we obtain a new graph through pruning that has only 14 process nodes, 18 state nodes, and 50 edges, as seen in Fig. 4.

From here, certain elements of the graph are grouped together. This is possible because there are always groups of rules that can be treated as functioning as a single rule. Equally, molecules (both in reality and in models) often have multiple phosphorylation sites that can be treated for the purpose of the regulatory graph as a single phosphorylation *state*. Through grouping, the program takes Fig. 4 and turns it into Fig. 5.

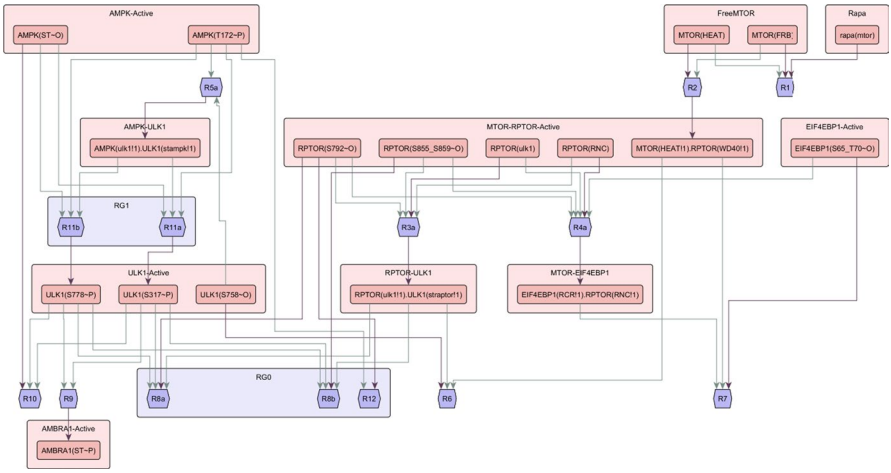

Fig. 5 Regulatory graph with elements grouped

This graph is then "collapsed." The collapse function is again mathematically defined, but intuitively, what it does is allow us to treat an entire group of nodes as a single entity (whose input/output function is the same as that of the entire group when taken as a whole), only when this does not affect the overall functioning of the graph. Applying this to Fig. 4, the outcome is a graph with 11 process nodes, 10 state nodes, and 33 edges, as depicted in Fig. 6.

Finally, the graph is annotated with the labels and diagrammatic conventions customary for biologists, including traditional arrow types and colours (Fig. 7).

As a reminder, the purpose of this exercise was not to see if a computer could recreate Fig. 1. The task performed by the computational systems biologists was to take Fig. 1 and use it to create a computer model capable of accurately simulating and predicting real empirical data. This they did. What the new piece of software does is create a visualization of the inner workings of that computer model. To see that progress has been made, note that there are important differences between Figs. 1 and 7, including additional arrows in Fig. 7 that do not exist in Fig. 1. These arrows represent previously unknown signalling pathways in the system. When investigated empirically, these pathways were found to be present in the target system. The computer model therefore contains novel information about the actual system's behaviour, and the visualization program helps collaborating biologists understand that new information. We take this to be a genuine scientific and epistemological achievement, both in the production of new knowledge and in the reduction of the kind of epistemic opacity that would have limited how well the computer model and its results could be understood by collaborators.

We now turn to a philosophical discussion of how this automated visualization program produces new understanding.

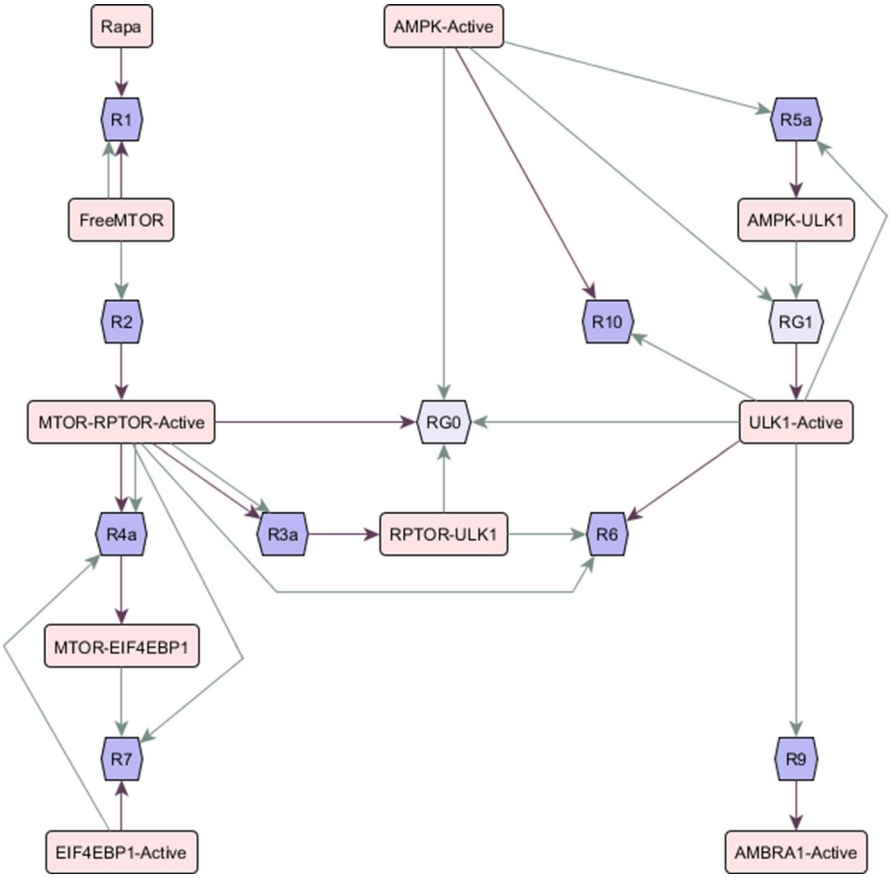

Fig. 6 Collapsed regulatory graph

4 Peeking Inside the Black Box

The automated model visualization program described above reduces epistemic opacity by opening up epistemic access to the model. But how does it open epistemic access, and how does that access lead to new understanding? We find two accounts in the literature helpful here. Their combination, we think, explains how the visualization achieves what it does. The first is Catherine Elgin's account of exemplification. The second is Kendall Walton's account of photographic snapshots.

For Elgin, exemplars are objects that simultaneously represent some features of a target while instantiating those features. An example is a sample of fabric, which *represents* the colour and texture of a larger piece of fabric that you might want to buy and make into, e.g., an article of clothing. At the same time, the fabric sample *is* the colour and texture of the unseen fabric it represents. That is, it instantiates those features of the fabric. According to Elgin, many inferences (scientific and otherwise) increase understanding by means of exemplification. Laboratory

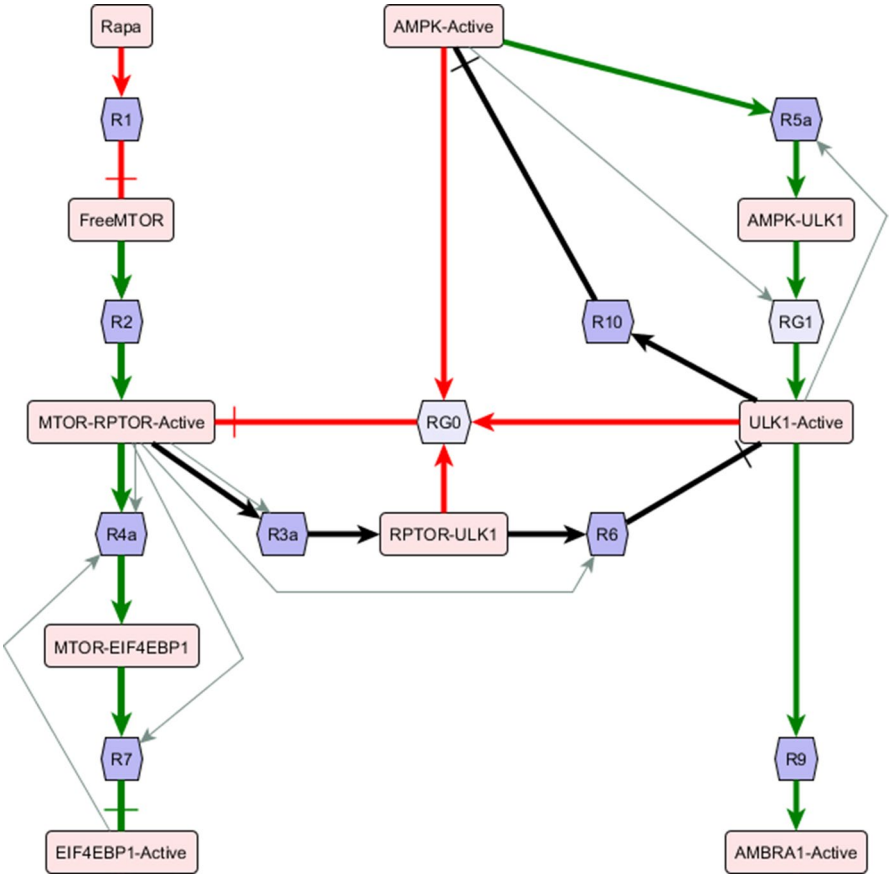

Fig. 7 Regulatory graph converted into standard conventional format

experiments, for example, are performed on artificially constructed systems that represent real world systems while also instantiating many features of the represented, real world systems. For example, the Miller–Urey experiment introduced electrical currents to pure samples of chemical elements contained in a sealed glass container. By this method, certain amino acids which we take to be important building blocks for life were created (Elgin 2014). The chemical elements in the containers were all thought to be present on prebiotic Earth, and the electrical discharges represented lightning, which was also present on prebiotic Earth. On interpretation, this achievement represents any and all instances in the history of the Earth where those same elements and events produced those particular amino acids. But the experiment doesn't merely *represent* cases where certain elements and events produced certain proteins, it actually *is* a case where certain elements and events produced certain proteins. In other words, it *instantiates* features of any such events on Earth, features such as containing certain elements combined in certain ways and producing amino acids. This provides epistemic access to those features. Through exemplification we

can come to understand one way life may have begun on Earth by understanding how certain elements led to certain proteins in the lab, just like we learn about the softness of a piece of fabric we've never seen by feeling a sample of it. Instantiation teaches us about certain features, representation transfers that new understanding to the target.

Thought experiments are another kind of scientific inference that can proceed by exemplification (Elgin 2014). Here, however, not all the relevant features can be instantiated. While the nitrogen in the Miller–Urey experiment really is nitrogen (and therefore has all the material features of real nitrogen), the nitrogen in our minds is not really nitrogen (and therefore does not have all the material features of real nitrogen). Thus while imaginings can represent nitrogen, they cannot instantiate all the features of nitrogen. So how do we learn from thought experiments and computer simulations, which cannot instantiate all the features of their target systems? Elgin's reply is that even if such representations cannot instantiate all the features of their target systems, they can still instantiate some of them, and these might be the relevant ones for inquiry. For example, the nitrogen atoms in our minds do not literally have 7 protons in their nuclei because the nitrogen atoms are thought-tokens, and thought-tokens are either immaterial (and so, are not made of protons) or they are patterns of firing among neurons (and so they are "made of" the wrong number of protons). But in a different sense, the nitrogen atoms in our minds *do* all have 7 protons in their nuclei because if they had some other number of protons, we would be imagining a different element. Real nitrogen atoms are composed of nuclei that have 7 protons, while the nitrogen atoms in our minds are also composed (in thought) of nuclei that have 7 protons. Thus "having 7 protons" is a feature instantiated by both real nitrogen atoms and by our mental representations of nitrogen atoms. Just as Sydney Carton in *A Tale of Two Cities* really is altruistic, even though he is fictional, the nitrogen atoms in our mental models really do have 7 protons (Elgin 2014, 228–229).[4] This applies to the kind of visualization presented above because the epistemologically relevant features can be found in both the model and the visualization. For example, the relationship between mTORC1 and ULK1 is described by the biochemical notion of mutual inhibition, in which two things reduce the activity of one another. Tokens of mTORC1 and ULK1 are related by mutual inhibition in the real system, but also in the computer model and diagram because in all three cases that is the relationship that holds between tokens of mTORC1 and ULK1. The tokens of mTORC1 and ULK1 in the computer model can be "active" or "inactive," and they regulate each other's activity by mutual inhibition. In the diagram, conventions of representation tell us that elements like mTORC1 and ULK1 are always active unless inhibited, and in this diagram, these two components of the pathway mutually inhibit one another. Thus, while the

[4] Elgin sometimes uses a different strategy that may amount to the same thing, using the notion of "metaphorical exemplification," that is, non-literal exemplification. Thus, a lifeless painting can instantiate optimism, and a mathematical proof can instantiate elegance (Elgin 2002). The painting made of canvass and paint has no feelings, so it is not literally optimistic. But we agree that it's an optimistic painting, so it instantiates optimism "metaphorically." This argument depends on considerations about the difference between what is metaphorical and what is not, which we will not go into here.

molecules we find in the model, diagram and real world are not "made of the same stuff" and will not have every feature in common, the tokens of mTORC1 stand in the same relation of mutual inhibition to tokens of ULK1 in all three cases.

Exemplification provides epistemic access because instantiating feature *p* is having feature *p*, and while we may not have direct cognitive access to all of the important features of the computer model, we do have access to the visualization of the model, which has and represents certain important features. This provides us with access to those features, which is important for increasing our understanding. But there is a problem. Many of the exemplars we've discussed so far are what might be called "established" exemplars (EEs). That is, they are all created in the following way: we want to help others understand some features of a system, so we create an exemplar that has and represents these features. The fabric sample has certain features people are interested in when buying fabric (colour and texture), and we can ensure that those features instantiated in the sample are the same as those in the fabric in the warehouse, e.g., by cutting the sample from the fabric. In our case, however, we have what might be called a "potential" exemplar (PE). In a PE, the visualization instantiates certain features and represents the computer model as having those features, but it isn't clear if those features really are present in the computer model. The reason for this is that we did not create the visualization as a way to explain something we already understood. Instead, we created it to give ourselves understanding of something we do not currently understand (the computer model). We thus need a way of guaranteeing that the feeling of understanding we get is a mark of genuine understanding.

One way to get such a guarantee is by external validation. For example, suppose a doctor performs an X-ray on your foot. The X-ray shows that your toe is broken. One way you can be sure that the X-ray accurately represents the state of your foot is by cutting your foot open and checking to see if the bone is actually broken just as the X-ray indicates. After having gone to this great effort, the PE becomes an EE because we now know that the same features (e.g., a break two-thirds of the way down the phalange) are instantiated in both the real system (your foot) and the exemplar (the X-ray).

But not all cases require external validation. Even for the X-ray, there are good reasons to think that the image exemplifies the brokenness of your toe without having to cut open your foot and look at your bones. One reason, in this case, comes from our theoretical knowledge of X-ray machines. If your toe was not broken, the X-ray image would not (typically) show a broken toe. The features of the exemplar are counterfactually dependent on the state of your foot. Call such exemplars *trustworthy* potential exemplars (TPEs). They are not established when they are not externally validated, but there is nonetheless reason to believe they would be validated if checked against the target system, so they are better than mere PEs.

Sometimes, external validation is not possible. For example, dinosaur bones in a museum represent and instantiate certain features (e.g., tooth length) of a creature that lived a long time ago. We cannot go back in time to verify that the dinosaur whose bones we are looking at now had teeth of that length. But we don't have to, because the teeth we are looking at now, while different (due to natural processes like material degradation) from the teeth the creature had at the time of its death,

are counterfactually dependent on the teeth of the living dinosaur. If the living dinosaur had shorter teeth, the teeth we are looking at now would be shorter. The dinosaur skeleton is thus also a TPE for some features of the living dinosaur that it represents. What is crucial here is the possibility of exemplars that cannot be established through external validation, but still stand as a trustworthy guide to their targets. TPEs allow us to understand *new* things about the system, without the need for external validation.

Thus, when we have a TPE, we can be relatively confident that it instantiates the relevant features of the target system, and this opens epistemic access to the target. At this stage, we can possess explanations of the model in terms of those features, successfully manipulate the model by learning to manipulate those features, and grasp the relations between the features. In other words, all three kinds of understanding can be gained through this sort of epistemic access, *if* the diagram is a TPE and not a mere PE. Since one thing that can make a PE into a TPE is counterfactual dependence of the exemplar on the target, we should discuss in more detail how this might obtain for visualizations in particular. A good start is Kendall Walton's account of snapshots.

Snapshots are photographs that intend to depict the objects in a photograph, while at the same time standing in an epistemically privileged (because causal) relation to those objects. It is thus a two-pronged account, like Elgin's exemplification. A photograph of Judy Garland *depicts* Judy Garland, in the sense that it is meant to "induce viewers to imagine" that they are seeing Judy Garland. It tells us imagine that our visual experience of the picture is a visual experience of Garland (Walton 2013). But the picture also stands in a *photographic* relation to Judy Garland. "The photographic relation is a causal relation of a certain kind. It has nothing essentially to do with viewers' experiences, and it does not have a normative dimension like that of depiction" (ibid). Counterfactual dependence is important to the photographic relation. If Judy Garland wasn't smiling when the shutter snapped, the photo would look different. This counterfactual dependence justifies inferences about the subject of the photograph, and this is why untampered photos (and videos, etc.) are admissible as evidence in court: they are counterfactually dependent on their objects, which is evidence that the objects depicted were as they appear to be in the photograph. Snapshots are photographs that depict an object (induce us to imagine it) while at the same time being counterfactually dependent on the features of that object.

The scientific visualization described in Sect. 3 is a snapshot in the sense that it is a depiction of the elements and dynamics of a computer model, and it is also strongly counterfactually dependent on that model. Being a depiction, we are invited to imagine that we are looking at the elements and dynamics of the model. In addition, if the computer model were relevantly different, the resulting visualization would also be different. This counterfactual dependence is not causal, but algorithmic. We see no reason to limit the kind of trustworthy counterfactual dependence to causal dependence; what matters is that the snapshot carries information about its target via counterfactual dependence.

The strength of the counterfactual dependence matters. In the case discussed in this paper, the counterfactual dependence is very strong. Unlike a camera, which

relies on light moving through a medium, there is nothing "between" the model and the diagram, and so there is nothing to obscure the sensitivity of the visualization to changes in the target system. The probability of misidentifying noise as signal (or vice versa) is very low. And the reliability of the process that creates the visualization from the model can be checked against simple models for which epistemic opacity is not an issue. That is, we can build a very simple model we understand well, and then run the algorithm and create a visualization, and ensure that it coheres with what we know to be true of the model. External validation is possible in these cases, and we confirm that the PE is actually an EE by comparing the visualization directly to the model. This increases our trust in the algorithm for cases in which we cannot provide this kind of external validation. Finally, unlike a camera, we can run the program backwards on any visualization it produces and retrieve the original model. This gives us another way to verify the accuracy of the process. So the counterfactual dependence is strong, the reliability of the method can be checked in two directions, and users of the diagram are in a position to know all of this, which contributes to the algorithm's epistemological trustworthiness.

These considerations justify our confidence in the visualization as a source of information about the elements and dynamics of computer models in computational systems biology. And this is often what we want from representations in science: to provide reliable guides to features of target systems. It suggests that what we learn about the relevant features exemplified by the diagram will be true of the computer model. For example, when biologists noticed the extra signalling pathways in Fig. 7 that were not present in Fig. 1, they gained evidence that the computer model also included such signalling pathways. This enabled the scientists to possess explanations of the model's output and behaviour in terms of those pathways (explanatory understanding); to gain new abilities concerning those pathways, like the abilities to produce new explanations and predictions about the model's behaviour, and to engage in meaningful conversation about the model that would not have previously been possible (manipulability understanding); and to grasp the coherence-making connections among aspects of the model's output and behaviour, e.g., the connections among those new pathways, other pathways, and the molecules in the model (objectual understanding). Thus, through exemplification and counterfactual dependence, the visualization is able to provide understanding and reduce the epistemic opacity of computer models in systems biology. Because understanding requires epistemic access, and epistemic opacity bars this access, restoring that access can increase understanding.[5]

[5] Of course, new knowledge might be produced as well. For example, the diagram can provide warrant for claims about the existence of those new pathways in the model's target system since (a) the diagram is counterfactually dependent on the computer model and (b) we have independent evidence that the model is accurate, so we can infer that this feature of the model is at least plausibly also instantiated in reality. But even in cases where no new knowledge is produced (e.g., there are no new pathways), we can still gain new understanding of the computational model through the diagram.

5 Conclusion

The kind of algorithmically generated model-visualization described in this paper can be used to address the problem of epistemic opacity for black box models of large signalling networks. We can expect more visualizations of this type to be created and used with other kinds of computational models in systems biology. And insofar as other scientific domains use relevantly similar models, perhaps algorithms like this one could also be helpful in reducing epistemic opacity in those domains.

One question to ask concerns the qualities that make certain instances of such diagrams better than others. According to P4, "the end goal would be to make an image, show it to a biologist, and they think you drew it by hand." Achieving this requires more than mere counterfactual dependence: it requires knowing what features of diagrams best foster human understanding. As P4 notes, however, understanding is "a little wishy-washy." "What you're trying to do is you're trying to promote understanding. There's not really a theory of how to promote understanding. Especially in the visual thing" (P4, interview, 02/22/2016). Philosophers are working on understanding scientific understanding and how diagrams appeal to imagination to produce it (see e.g., Baumberger forthcoming; de Regt 2009, 2017; Elgin 2017; Meynell 2018; Stuart 2016, 2017, 2018). While we wait for such accounts to be fleshed out, however, there are still things we can say now. For example, effective diagrams employ space, colour and dimension in ways that appeal to basic human intuition (see, e.g., de Regt 2014; Gansterer 2011; Meynell 2018; Nersessian 2008; Tufte 2001). In addition, they portray their targets using conventions that are familiar (or could become familiar) to the epistemic community. And while diagrams can exemplify many different features, they should exemplify those features of the model that are important for giving explanations, gaining new abilities, and grasping the connections between the elements of the model.

In sum, our investigation into the practices of this computational systems biology lab has shown that scientists can eliminate some of the epistemic opacity that accompanies computer models without finding a way to lay bare all the inferential steps made therein. They can unlock epistemic access to significant features of computer models, including their elements and dynamics, by means of an image. This counts, we think, as a solution to one version of the problem of epistemic opacity of computer models in science.

Acknowledgements We would like to thank the Center for Philosophy of Science of the University of Pittsburgh for funding while carrying out this research. Mike Stuart thanks the Social Sciences and Humanities Research Council of Canada for funding and the Centre for Philosophy of Natural and Social Science at the London School of Economics, and especially Roman Frigg, for support. We also thank the Lab Director and researchers in our study for welcoming us into their lab and granting us numerous interviews. For feedback we would like to thank Rami El Ali, Chiara Ambrosio, Agnes Bolinska, Hasok Chang, Johannes Lenhard, Josh Norton, Jacob Stegenga, Adam Toon, and two anonymous reviewers of this paper, as well as audiences at the Society for Philosophy of Science in Practice at the University of Ghent, the Lebanese American University, the UK Integrated HPS workshop in Nottingham, and the Imagination in Science Conference at the University of Leeds.

References

Baumberger, C. (2011). Types of understanding: Their nature and their relation to knowledge. *Conceptus, 40,* 67–88.

Baumberger, C. (Forthcoming). Explicating objectual understanding taking degrees seriously. *Journal for General Philosophy of Science*.

Baumberger, C., & Brun, G. (2016). Dimensions of objectual understanding. In S. Grimm, C. Baumberger, & S. Ammon (Eds.), *Explaining understanding: New essays in epistemology and the philosophy of science*. London: Routledge.

Chandrasekharan, S., & Nersessian, N. J. (2015). Building cognition: The construction of external representations for discovery. *Cognitive Science, 39,* 1727–1763.

de Regt, H. (2009). Understanding and scientific explanation. In H. De Regt, S. Leonelli, & K. Eigner (Eds.), *Scientific understanding*. Pittsburgh: University of Pittsburgh Press.

de Regt, H. (2014). Visualization as a tool for understanding. *Perspectives on Science, 22,* 377–396.

de Regt, H. (2017). Understanding scientific understanding. Oxford: Oxford University Press.

Dellsén, F. (2018). Beyond explanation: Understanding as dependency modelling. *British Journal for the Philosophy of Science* axy058. https://doi.org/10.1093/bjps/axy058.

Elgin, C. Z. (2002). Art in the advancement of understanding. *American Philosophical Quarterly, 39,* 1–12.

Elgin, C. Z. (2007). Understanding and the facts. *Philosophical Studies, 132,* 33–42.

Elgin, C. Z. (2011). Making manifest: The role of exemplification in the sciences and the arts. *Principia: An International Journal of Epistemology, 15,* 399–413.

Elgin, C. Z. (2014). Fiction as thought experiment. *Perspectives on Science, 22,* 221–241.

Elgin, C. Z. (2017). *True enough*. Cambridge: MIT Press.

Gansterer, N. (Ed.). (2011). *Drawing a hypothesis*. Vienna: Springer.

Grimm, S. (2008). Epistemic goals and epistemic values. *Philosophy and Phenomenological Research, 77,* 725–744.

Hannon, M. (forthcoming). What's the point of understanding?

Hempel, C. G. (1965). *Aspects of scientific explanation and other essays in the philosophy of science*. New York: The Free Press.

Hills, A. (2015). Understanding why. *Nous, 50,* 661–688. https://doi.org/10.1111/nous.12092.

Humphreys, P. (2004). *Extending ourselves: Computational science, empiricism, and scientific method*. Oxford: Oxford University Press.

Humphreys, P. (2009). The philosophical novelty of computer simulation methods. *Synthese, 169,* 615–626.

Kelp, C. (2015). Understanding phenomena. *Synthese, 192,* 3799–3816.

Khalifa, K. (2012). Inaugurating understanding or repackaging explanation? *Philosophy of Science, 79,* 15–37.

Khalifa, K. (2013). Is understanding explanatory or objectual? *Synthese, 190*, 1153–1171.

Kitcher, P. (1989). Explanatory unification and the causal structure of the world. In P. Kitcher & W. Salmon (Eds.), *Scientific explanation*. Minneapolis: University of Minnesota Press.

Kvanvig, J. (2009). The value of understanding. In A. Haddock, A. Millar, & D. Pritchard (Eds.), *Epistemic value*. Oxford: Oxford University Press.

Lenhard, J. (2006). Surprised by a nanowire: Simulation, control, and understanding. *Philosophy of Science, 73*(5), 605–616.

Lenhard, J. (2018). Thought experiments and simulation experiments: Exploring hypothetical worlds. In M. Stuart, et al. (Eds.), *The Routledge companion to thought experiments* (pp. 484–497). London: Routledge.

MacLeod, M., & Nersessian, N. J. (2016). Interdisciplinary problem solving: emerging modes in integrative systems biology. *European Journal for the Philosophy of Science, 7/16(6),* 401–418.

Meynell, L. (2018). Images and imagination in thought experiments. In M. Stuart, et al. (Eds.), *The Routledge companion to thought experiments* (pp. 498–511). London: Routledge.

Mizushima, N., & Komatsu, M. (2011). Autophagy: Renovation of cells and tissues. *Cell, 147,* 728–741.

Nersessian, N. J. (2008). *Creating scientific concepts*. Cambridge, MA: MIT Press.

Parker, W. (2014). Computer simulation. In M. Curd & S. Psillos (Eds.), *The Routledge companion to philosophy of science* (pp. 136–146). London: Routledge.

Pritchard, D. (2010). Knowledge and understanding. In A. Haddock, A. Millar, & D. Pritchard (Eds.), *The nature and value of knowledge: Three investigations*. Oxford: Oxford University Press.

Strevens, M. (2013). No understanding without explanation. *Studies in History and Philosophy of Science, 44,* 510–515.

Stuart, M. T. (2016). Taming theory with thought experiments: Understanding and scientific progress. *Studies in the History and Philosophy of Science, 58,* 24–33.

Stuart, M. T. (2017). Imagination: A sine qua non of science. *Croatian Journal of Philosophy, XVII*(49), 9–32.

Stuart, M. T. (2018). How thought experiments increase understanding. In M. Stuart, et al. (Eds.), *The Routledge companion to thought experiments* (pp. 526–544). London: Routledge.

Szymańska, P., Martin, K. R., MacKeigan, J. P., Hlavacek, W. S., & Lipniacki, T. (2015). Computational analysis of an autophagy/translation switch based on mutual inhibition of MTORC1 and ULK1. *PLoS ONE, 10*(3), e0116550. https://doi.org/10.1371/journal.pone.0116550.

Tufte, E. (2001). *The visual display of quantitative information*. Cheshire, CT: Graphics Press.

Walton, K. (1984). Transparent pictures: On the nature of photographic realism. *Critical Inquiry, 11,* 246–277.

Walton, K. (2013). Fotografische Bilder. In J. Nida-Rümelin & J. Steinbrenner (Eds.), *Fotografiezwischen Dokmentation und Inszenierung* (pp. 11–28). Berlin: Hatje Cantz Verlag.

Wilkenfeld, D. A. (2013). Understanding as representation manipulability. *Synthese, 190,* 997–1016.

Wilkenfeld, D. A. (2014). Functional explaining: A new approach to the philosophy of explanation. Synthese, 191, 3367–3391.

Wilkenfeld, D. A. (2017). MUDdy understanding. *Synthese, 194,* 1273–1293.

Wilkenfeld, D. A., & Hellmann, J. K. (2014). Understanding beyond grasping propositions: A discussion of chess and fish. *Studies in the History and Philosophy of Science, 48,* 46–51.

Wilson, R. (2014/2002). *Four Colors Suffice*. Princeton: Princeton Science Library, Princeton University Press.